RECEIVED
CENTRAL FAX CENTER
DEC 07 2006

REMARKS

In the above-identified Office Action, the Examiner rejected Claims 1 and 23 - 39 under 35 U.S.C. §103(a) as being unpatentable over Pajak et al. in view of Bartram et al.

Applicants have amended Claims 1, 23, 25, 26, 28, 29, 31, 32, 34, 35, 37 and 38 to better claim the invention. Claims 1 and 23 – 39 remain pending in the Application. For the reasons stated more fully below, Applicants submit that the claims are allowable over the applied references. Hence, reconsideration, allowance and passage to issue are respectfully requested.

As stated in the Responses to the Previous Office Actions as well as disclosed in the SPECIFICATION, in a typical distributed data processing system, there are many situations where a user may need to display and interact with both local and remote data objects. Local and remote data objects may include, for example, files and folders located locally and remotely. Typical actions performed on these local and remote data objects may include, for example, editing, deleting, copying, moving, renaming, and in the case of program files and executable files, compiling and running.

Known data object viewers typically display either all local or all remote data objects in a single view. When interacting with remote data objects presented in such a view, two approaches have been commonly used. In a first approach, an action on a remote data object is invoked directly on the remote system, for example, by programming a customized remote action. In a second approach, a remote data object is "pulled" or downloaded to a local system, acted upon locally, and then "pushed" or uploaded back to the remote system. For example, there are Windows-based code editors, which allow for local editing of remote files. Typically, this is done by downloading a temporary copy of the remote files to a local system, allowing the files to be edited locally, and then subsequently pushing the edited files back to the remote system. Thus, there is

a need to provide a more flexible approach to displaying and interacting with local and remote data objects.

The present invention provides such a flexible approach. In accordance with the teachings of the invention, both data objects stored locally and remotely may be displayed in a viewer. In the case where a particular data object is stored both locally and remotely, the data object may be displayed as a hybrid data object that represents both the locally stored data object and the remotely stored data object. When a user selects a displayed hybrid data object on which to perform an action, the actual data object on which the action is to be performed will be requested (i.e., whether the action is to be performed on the locally stored data object only, the remotely stored data object only or on both the locally stored and remotely stored data objects). Once the user indicates on which data object the action is to be performed, it is done.

The invention is set forth in claims of varying scopes of which Claim 1 is illustrative.

> 1. A method of interacting with locally and remotely stored data objects in a distributed data processing system, comprising:
>
> determining whether a data object is stored on both a remote system in the distributed data processing system and a local system;
>
> ***displaying on the local system, if it is determined that the data object is stored on both the local system and the remote system in the distributed data processing system, the data object as a hybrid data object, the hybrid data object representing both the data object stored on the local system and the data object stored on the remote system;***
>
> ***enabling a user on the local system to perform an action on the hybrid data object by first selecting the hybrid data object;***
>
> ***prompting the user, in response to the user selecting the hybrid data object, to indicate whether the action is to be performed on the data object stored on the local system, the data object stored on the remote system, or both the data object stored on***

> ***the local system and the data object stored on the remote system***; and
> performing the action as indicated by the user. (Emphasis added.)

Applicants submit that the claims are allowable over the applied references.

Pajak et al. purport to teach a multi-user collaborative system. According to the purported teachings of Pajak et al., different users at different workstations connected to a common link may access shared structured data objects or files. However, editing or modification of a shared file cannot be performed by a user unless the user has first locked out all other users from editing the file. Specifically, when the user wants to open a shared file for editing, the user has to lock the file from the other users before opening the file; otherwise, the file is opened in a read-only mode. When the file is opened, a copy of the file is downloaded to the user's workstation in order to be displayed thereon (see col. 16, lines 61 – 64 and col. 21, line 43 to col. 22, line 5).

Visual indication as to the state of a shared file (i.e., whether or not the shared file is locked) and other information relating to a user that locks the file including the time the file was locked are updated and displayed in a window at a user's workstation. For example, a small terminal icon in a shared book window that is displayed on a user's workstation signifies that an open file has been modified locally by a user; a black lock indicates that it is locked by the user (a gray lock indicates that it is locked by another user) and a plus sign indicates that the local displayed copy is more recent than the copy on the server. Thus, through the shared book window any user may determine the exact status of all the shared files.

But note that a hybrid data object is not displayed on the local system as claimed. What is displayed on the local system of Pajak et al. is an indication as to whether the locally displayed file is more or less recent than the remote copy of the file. Thus, the displayed data object (e.g., name of the file with a "plus"

sign) is a representation of only the locally displayed data object. Therefore, Pajak et al. do not teach the step of ***displaying on the local system, if it is determined that the data object is stored on both the local system and the remote system in the distributed data processing system, the data object as a hybrid data object, the hybrid data object representing both the data object stored on the local system and the data object stored on the remote system***.

And since Pajak et al. do not teach a hybrid data object, they, therefore, cannot teach the steps of ***enabling a user on the local system to perform an action on the hybrid data object by first selecting the hybrid data object***; and ***prompting the user, in response to the user selecting the hybrid data object, to indicate whether the action is to be performed on the data object stored on the local system, the data object stored on the remote system, or both the data object stored on the local system and the data object stored on the remote system***.

The Examiner asserted that Pajak et al. ***clearly*** teach the prompting step. However, the Examiner is unable to cite any passage in the disclosure of Pajak et al. where the prompting step is succinctly disclosed. Applicants submit that Pajak et al. do not teach, show or so much as suggest such step.

Pajak et al. specifically teach that when a user is opening a file locally, it is the remote copy of the file that is downloaded, opened and displayed locally (see col. 21, lines 51 – 62). When the user modifies the displayed copy of the file (locally) and then save the file, it is the remote copy of the file that is automatically replaced by the locally-modified display of the file (see col. 22, lines 6 – 42). Thus, since the remote copy of the file is always the one that is being acted upon, there is no reason for Pajak et al. to teach the step of ***prompting the user, in response to the user selecting the hybrid data object, to indicate whether the action is to be performed on the data object stored on the local system, the data object stored on the remote system, or both the data***

object stored on the local system and the data object stored on the remote system and in fact as submitted above, Pajak et al. do not teach such step.

Bartram et al. purport to teach a system and method for distributing shared storage for collaboration across multiple devices. According to the teachings of Bartram et al., objects can be shared across a number of units on a peer-to-peer network. The objects are supplied by individual users and reflected at other units in a user interface view dedicated to a shared store using object anchors. The object anchors are pointers to the objects and not the object itself. Users can make a copy of any remote object by selecting the copy operation on the remote object's anchor. Multiple local copies can therefore exist on the shared store and are indicated by the object's owners to others. When a person's local copy needs to be merged or reconciled with another, the version information can be supplied to the appropriate people and the decision of how to synchronize a file content is left to the user's specific application tools. By using anchors rather than objects to continuously exchange an update across the shared store, the bandwidth and local storage requirement for the units in the peer-to-peer network are reduced. Additionally, a large collaborative network with a large shared store can be represented with a very small physical storage capacity, such as those used with PDAs or mobile phones, since not all the objects need to be stored by every peer but only the anchors which point to the objects.

However, just as in the case of Pajak et al., Bartram et al. do not teach the steps of (1) ***displaying on the local system, if it is determined that the data object is stored on both the local system and the remote system in the distributed data processing system, the data object as a hybrid data object, the hybrid data object representing both the data object stored on the local system and the data object stored on the remote system***; (2) ***enabling a user on the local system to perform an action on the hybrid data object by first selecting the hybrid data object***; and (3) ***prompting the user, in response to the user selecting the hybrid data object, to indicate whether***

DEC 07 2006

the action is to be performed on the data object stored on the local system, the data object stored on the remote system, or both the data object stored on the local system and the data object stored on the remote system.

Since neither Pajak et al. nor Bartram et al. teach the emboldened/italicized limitations in the above-reproduced Claim 1, combining the teachings of the Pajak et al. with those of Bartram et al. does not show, teach or suggest the claimed invention.

Consequently, Applicants submit that Claim 1, as well as its dependent claims, should be allowable. Independent Claims 28 and 34, which all incorporate the above-emboldened-italicized limitations in the above-reproduced claim 1, together with their dependent claims, should also be allowable. Hence, Applicants once more respectfully request reconsideration, allowance and passage to issue of the claims in the application.

Respectfully Submitted

By: _____
Volel Emile
Attorney for Applicants
Registration No. 39,969
(512) 306-7969